B: Amendments to The Claims:

What is claimed is:

Claim 1. (Cancelled)

Claim 2. (Cancelled)

Claim 3. (Cancelled)

Claim 4. (Cancelled)

Claim 5. (Cancelled)

Claim 6. (Cancelled)

Claim 7. (Cancelled)

Claim 8. (Cancelled)

Claim 9. (Cancelled)

Claim 10. (Cancelled)

Claim 11. (Cancelled)

Claim 12. (Cancelled)

Claim 13. (Cancelled)

Claim 14. (Cancelled)

Claim 15. (Currently amended) **A branch prediction process for a computer system for improving branch prediction rate when using a branch history table, ~~as defined by claim 14, further~~ comprising:**

determining if a program instruction processor (processor) has an access hit (hit) or access miss (miss) in an instruction cache (I-cache) when utilizing an instruction address (IFAR address) in attempting to select a program instruction for execution by the processor,

generating a hint instruction when the instruction is a branch in response to a hit occurring during the determining operation, storing the hint instruction in association with a copy of an instruction line containing the program instruction in a storage hierarchy of the computer system, the hint instruction storing BHT prediction fields obtained from a copy of a current BHT entry associated with the program nstruction line when the hit occurs, and storing a branch mask in the hint

instruction for locating an associated BHT field indicating the BHT field associated with the location of the instruction in the instruction line, and

transferring the copy of the instruction line and associated hint instruction from the storage hierarchy to the I-cache in response to a miss occurring during the determining operation, and executing the hint instruction to restore a BHT prediction field in a current BHT entry to the state of a BHT field in the hint instruction located by the branch mask, and

the generating operation of generating hint instructions being performed by a hint processor operating in parallel with the program instruction processor, and

executing a hint instruction when the hint instruction is received in the I-cache by testing an operation code field in the hint instruction to determine if a completed hint instruction is indicated or if a no-operation state is indicated for the hint instruction, and continuing the executing process only if a completed hint instruction is indicated by performing the following operations:

reading a BHT entry in the BHT located at an index determined by a bht_index field in the hint instruction, and storing the BHT entry in a curr_bht register,

logically ANDing an Nth bit in an inversion of the branch_mask field in the hint instruction with an Nth bit in the curr_bht register, where N is the bit position of the current instruction in the instruction line, and logically ANDing the Nth bit in a branch_mask field with an Nth bit in a bht_bits in the hint instruction,

logically ORing outputs of the two logical ANDing operations to provide an Nth bit output, and setting an Nth bit in a new_bht register to the Nth bit output,

receiving without change in the new_bht register at bit locations other than at the Nth bit location the bits in the curr_bht register at corresponding bit locations other than the Nth location, and

setting the content of the new_bht register into the current BHT entry in the BHT to restore the BHT entry to its last prediction state for the current instruction.

16. (Currently Amended) The [A] branch prediction process for a computer system for improving branch prediction rates when using a branch history table as defined by claim 15, further comprising:

performing all of the hint instruction operations in a hint instruction processor.

17. (Currently Amended) The [A] branch prediction process for a computer system for improving branch prediction rates when using a branch history table as defined by claim 16, further comprising:

performing all hint instruction operations and all program instruction processor operations in a single semiconductor chip.